Status of Proposed CF Conventions for Point Observation Data

John Caron

Oct 06, 2009

What we already have in CF

- Standard names, units, coordinate identification, etc…
- General Coordinate Systems
  - dataVar:coordinates = “lat lon height time”;
- Two specific examples
  - 5.4: Time Series of Station Data
  - 5.5: Trajectories

5.5: Trajectory

```
dimensions:
   time = 1000 ;
variables:
   float O3(time) ;
      O3:coordinates = "lon lat z time" ;
   double time(time) ;
   float lon(time) ;
   float lat(time) ;
   float z(time) ;
```

- Only one trajectory in the file
- Identical to collection of points (eg lightning)

5.4: Time Series of Station Data

```
dimensions:
   station = 10 ; // measurement locations
   pressure = 11 ; // pressure levels
   time = UNLIMITED ;
variables:
   float humidity(time, pressure, station) ;
      humidity:coordinates = "lon lat pressure time" ;
   double time(time) ;
   float lon(station) ;
   float lat(station) ;
   float pressure(pressure) ;
```

- Multidimensional representation: Same # of time steps at each station
- time(time) – same time coordinates at each station

Need to Add or Clarify

- Unambiguous categorization of data
- Ability to store one or many features in a file
- Correctly represent coordinate values, eg
  - Same/different time coordinates for all stations
  - Same/different z coordinates for all profiles
- Allow user control of space / time tradeoffs
  - Rectangular vs. ragged arrays
  - Unlimited dimension – append data
  - Group data on disk for optimal read pattern

CDM Design Goals

- Categorize point data into small number of types
- Accept current practices for storing point data (as much as possible without too much complication)
- Allow storing multiple features in one file
- Allow variable length features
- Allow user to decide on how much data redundancy
- Enable generic software to read files
- Enable aggregation of multiple files

Proposed Point data Categorization (aka “Feature Types”)

- **Point**: measured at one point in time and space
- **Station**: time-series of points at the same location
- **Profile**: points along a vertical line
- **Station Profile**: a time-series of profiles at same location.
- **Trajectory**: points along a 1D curve in time/space
- **Section**: a collection of profile features which originate along a trajectory.

CSML <-> CDM Feature types

(remarkably consistent)

CSML Feature Type	CDM Feature Type
PointFeature	PointFeature
PointSeriesFeature	StationFeature
TrajectoryFeature	TrajectoryFeature
PointCollectionFeature	StationFeature at fixed time
ProfileFeature	ProfileFeature
ProfileSeriesFeature	StationProfileFeature at one location and fixed vertical levels
RaggedProfileSeriesFeature	StationProfileFeature at one location
SectionFeature	SectionFeature with fixed number of vertical levels
RaggedSectionFeature	SectionFeature
ScanningRadarFeature	RadialFeature
GridFeature	GridFeature at a single time
GridSeriesFeature	GridFeature
SwathFeature	SwathFeature

Proposed Encoding Variations

- Rectangular Array
  - Multidimensional
  - Single : one feature in the file
- Ragged Array – different length features
  - Contiguous
  - Non-Contiguous
  - Flattened

Multidim

```
dimensions:
   traj= 21;
   obs = 777;
variables:
    float O3(traj, obs ) ;
    double time(traj, obs ) ;
    float lon(traj, obs ) ;
    float lat(traj, obs ) ;
    float z(traj, obs ) ;
```

Single

```
dimensions:
    time = 1000 ;
variables:
    float O3(time) ;
    double time(time) ;
    float lon(time) ;
    float lat(time) ;
    float z(time) ;
```

Ragged contiguous

```
dimensions:
   traj= 21;
   obs = 34509;
variables:
    traj_name(traj)
    traj_nobs(traj)
    float O3(obs ) ;
    double time(obs ) ;
    float lon(obs ) ;
    float lat(obs ) ;
    float z(obs ) ;
```

Ragged non-contiguous

```
dimensions:
   traj= 21;
   obs = 34509;
variables:
    traj_name(traj)
    traj_index(obs)
    float O3(obs ) ;
    double time(obs ) ;
    float lon(obs ) ;
    float lat(obs ) ;
    float z(obs ) ;
```

Ragged flattened

```
dimensions:
  obs = 34509;
variables:
    traj_name(obs)
    traj_desc(obs)
    float O3(obs ) ;
    double time(obs ) ;
    float lon(obs ) ;
    float lat(obs ) ;
    float z(obs ) ;
```

Status

- Proposed 1 year ago by John Caron
  - Steve Hankin is ticket moderator
- Still gathering use cases and sample files
  - 10-15 use cases (half from IDD, half from others)
  - All use cases (so far) can be satisfied
- Implemented in netCDF-Java library
  - New “Point Feature Dataset” APIs
  - Creating data query services in TDS
  - Testing local and remote access in IDV

Concerns

- Main problem is complexity
  - esp for stationProfile, section (double nested)
  - Recognizing all the variants in generic software is not trivial
- Not sure too many others have really read the whole thing
  - The usual CF problem
- Debating whether we need all these variations
  - Tentative conclusion : yes
  - But open to persuasion….

Conclusion

- I'd like to make one more pass at clarifying the proposal, now that Ive implemented much of it
- Then Id like to submit it and get on with it
- Any further feedback would be great NOW
- Sorry not to be in Hamburg :^(
- I'll tune in on the General CF Discussion
  - 14:30 hamburg, 6:30 colorado